Laboratory work 1

Sorbec Mihai, IS-211 fbf

Additional sorry for making this so lame-like. It is first time I see jupyter and it is 5 minutes since I saw markdowns. So far I don't even see what parts of the code should be commented more, so I am hoping I will get better at it.

In [152]:
```
# 1. imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [153]:
```
# 2. upload the dataset
dataset = pd.read_csv("diabetes.csv")
```

In [153]:
```
# 3. explore dataset
dataset.describe()
```

Out[153]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In third task there are obviously too many zeroes that showed up.

In [154]:
```
# 4. list variable names
dataset.columns.tolist()
```

Out[154]:
```
['Pregnancies',
 'Glucose',
 'BloodPressure',
 'SkinThickness',
 'Insulin',
 'BMI',
 'DiabetesPedigreeFunction',
 'Age',
 'Outcome']
```

0.5 pt

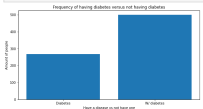
In [155]:
```
# 5. count how many have diabetes and how many don't
value_counts = dataset.Outcome.value_counts()
print("Diabetes counts:")
print(value_counts.head())
```

```
Diabetes counts:
0    500
1    268
Name: Outcome, dtype: int64
```

Do how many people have diabetes and how many don't?

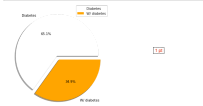
0.5 pt

In [156]:
```
# 6. Create a bar chart to represent the frequencies of those with diabetes versus those without diabetes,
# and label the axis and name the chart.
# 0 diabetes and 1 diabetes
diabetes = value_counts[0]
diabetes_barsen_labels = ["Diabetes", "No diabetes"]
plt.bar(Diabetes_barsen_labels, [diabetes, not_diabetes])
plt.title("Frequency of having diabetes versus not having diabetes")
plt.xlabel("Have a disease or not have one")
plt.ylabel("Amount of people")
plt.show()
```



1 pt

In [157]:
```
# 7. Create a pie plot, change the colors, add the labels, add shady, and "explode" the pie slice for those the
# Create the legend and put it into the right corner so that it does cover the pie plot.
# Name the legend "Diabetes"
plt.pie(value_counts, labels=diabetes_barsen_labels, colors=pie_colors, explode=[0, 0.1], shadow=True, autopct=
# matplotlib.inferno("text.color") / "white" if for dark theme..
plt.title("Diabetes")
plt.legend(loc="upper right")
plt.subplots_adjust(left=0.01, bottom=0.001, right=0.75)
plt.show()
```



1 pt

In [158]:
```
# 8. Calculate and report the mean the standard deviation of the BMI variable.
bmi_mean = dataset.BMI.mean()
bmi_std = dataset.BMI.std()
print("BMI mean: " + str(bmi_mean))
print("BMI standard deviation: " + str(bmi_std))
```

```
BMI mean: 31.992578124999998
BMI standard deviation: 7.8841603203754405
```

0.5 pt

In [159]:
```
# 9. By using the Chebyshev's Theorem infer what is the range for the BMI variable for 75% of the people in yo
# What is the range for 89% of the people in your dataset.
# 1) mean - (linear boundary) * within number
# 2) mean + (linear boundary) * within number
# k = number of standard deviations
# k = the within number / std
# m = mean, s = std, n = percentage
m75 = np.mean(2/(1 - 75/100)) * bmi_std
m89 = np.mean(2/(1 - 89/100)) * bmi_std
print(m75)
print("range for 75% is between " + str(bmi_mean - m75) + " and " + str(bmi_mean + m75))
print(m89)
print("range for 89% is between " + str(bmi_mean - m89) + " and " + str(bmi_mean + m89))
```

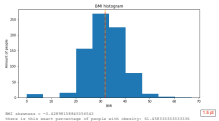
```
15.768320407508962
range for 75% is between 16.224257592429128 and 47.760897657571796
22.715277904325125
range for 89% is between 9.2229403324367672 and 55.762215915632332
```

0.5 pt

In [160]:
```
# 10. Create a histogram for the BMI variable, plot the mean and median on the histogram to answer the follow
# qu   In the BMI distribution skewed based on the position of the mean versus the median?
# bc    Calculate the skewness of BMI
# ir    Based on the value of BMI median and on the chart below infer what percentage of people in your dataset
# create histogram
plt.hist(dataset.BMI)
plt.title("BMI histogram")
plt.xlabel("BMI")
plt.ylabel("Amount of people")
bmi_median = dataset.BMI.median()
print("BMI median: " + str(bmi_median))
plt.axvline(bmi_mean, color="yellow", linestyle="-")
plt.axvline(bmi_median, color="red", linestyle="-")
plt.xticks(ticks=[*Figure.figsize*] * [16, 9])
plt.show()
# ir it doesn't mean that much skewed based on any of these  variables,
# because the mean and median are very close there are tails on the left side
bmi_skewness = dataset.BMI.skew()
print("BMI skewness = " + str(bmi_skewness))
# ir
# obese people are with bmi > 30 or higher
print("There is this smart percentage of people with obesity: " + str(len(dataset.BMI[dataset.BMI >= 30])/(lenth
```

Why variables you mean mean and median?

```
BMI median = 32.0
```

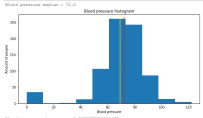



```
BMI skewness = -0.42898159420354143
There is this smart percentage of people with obesity: 41.458333333333336
```

0.5 pt

In [161]:
```
# 11. doing step 10 for Blood pressure
plt.hist(dataset.BloodPressure)
plt.title("Blood pressure histogram")
plt.xlabel("Blood pressure")
plt.ylabel("Amount of people")
bloodpressure_median = dataset.BloodPressure.median()
print("Blood pressure median = " + str(bloodpressure_median))
plt.axvline(bloodpressure_median, color="red", linestyle="-")
plt.axvline(bloodpressure_mean, color="yellow", linestyle="-")
plt.xticks(ticks=[*Figure.figsize*] * [16, 9])
plt.show()
# ir distribution is skewed  on the mean more it seems
# ir  Because the mean and median are very close and tails are on the left, but it is also VERY big, so we shoul
bloodpressure_skewness = dataset.BloodPressure.skew()
print("Blood pressure skewness = " + str(bloodpressure_skewness))
# ir
# density is not the thing here
# I found out that with more outliers on the left the mean value became less descriptive and according to the t
# mint to the median instead
```

```
Blood pressure median = 72.0
```



1.5 pt

```
Blood pressure skewness = -1.8436078140421522
```

In [162]:
```
# 12. Remove all the observations where the BloodPressure= and the BMI variables take the value of zero.
dataset = dataset.drop(dataset[dataset.BloodPressure == 0].index) # for some reason it stopped working in one
dataset = dataset.drop(dataset[dataset.BMI == 0].index)
dataset.describe()
```

0 pt

Out[162]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	729.000000	729.000000	729.000000	729.000000	729.000000	729.000000	729.000000	729.000000	729.000000
mean	3.838711	121.686638	72.387637	27.089114	80.948522	32.699919	0.474177	33.318244	0.344307
std	3.370348	32.250275	12.373826	15.738276	116.635505	6.885916	0.331481	11.753278	0.475438
min	0.000000	0.000000	24.000000	0.000000	0.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	64.000000	0.000000	0.000000	27.500000	0.243000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.400000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	33.000000	130.000000	36.600000	0.627000	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In [163]:
```
# 13-14. Now perform once again all the steps in 9, 10 and 11:
# 9. Calculate and report the mean the standard deviation of the BMI variable.
bmi_mean = dataset.BMI.mean()
bmi_std = dataset.BMI.std()
print("BMI mean: " + str(bmi_mean))
print("BMI standard deviation: " + str(bmi_std))
# mean is now bigger, std is now less
```

```
BMI mean: 32.498588271964
BMI standard deviation: 6.8850981882287
```

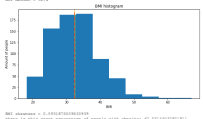
In [164]:
```
# 15.9. By using the Chebyshev's Theorem infer what is the range for the BMI variable for 75% of the people in
# What is the range for 89% of the people in your dataset.
# 1) mean - (linear boundary) * within number
# 2) mean + (linear boundary) * within number
# k = number of standard deviations
# k = the within number / std
# m = mean, s = std, n = percentage
m75 = np.mean(2/(1 - 75/100)) * bmi_std
m89 = np.mean(2/(1 - 89/100)) * bmi_std
print(m75)
print("range for 75% is between " + str(bmi_mean - m75) + " and " + str(bmi_mean + m75))
print(m89)
print("range for 89% is between " + str(bmi_mean - m89) + " and " + str(bmi_mean + m89))
# as a result either numbers are smaller, the range also became smaller because were seemed to be just outlie
```

```
13.770196270245762
range for 75% is between 18.699762271693893 and 46.297013272906224
20.661294405315
range for 89% is between 11.716047293253539 and 53.2801309649736
```

In [165]:
```
# 16.10. Create a histogram for the BMI variable, plot the mean and median on the histogram to answer the fo
# qu   In the BMI distribution skewed based on the position of the mean versus the median?
# bc    Calculate the skewness of BMI
# ir    Based on the value of BMI median and on the chart below infer what percentage of people in your dataset
# create histogram
plt.hist(dataset.BMI)
plt.title("BMI histogram")
plt.xlabel("BMI")
plt.ylabel("Amount of people")
bmi_median = dataset.BMI.median()
print("BMI median: " + str(bmi_median))
plt.axvline(bmi_mean, color="yellow", linestyle="-")
plt.axvline(bmi_median, color="red", linestyle="-")
plt.xticks(ticks=[*Figure.figsize*] * [16, 9])
plt.show()
bmi_skewness = dataset.BMI.skew()
print("BMI skewness = " + str(bmi_skewness))
# ir
# obese people are with bmi > 30 or higher
print("There is this smart percentage of people with obesity: " + str(len(dataset.BMI[dataset.BMI >= 30])/(lenth
# ir  I removed outliers and bigger, obesity without those seems also appears more often.
# mean and median also were closer
# how can obesseses be bigger when i removed the outliers?..
# also they are no longer on the highest bar, which is also weird for median I guess.
```

The outliers were values, they were not the people with obesity. Great remark, just it is not the thing to do right. Also the value became smaller, not bigger, mean and median, because of these values.

```
BMI median = 32.0
```



You should make your conclusions by looking at the graphs, not just by the just numbers the histogr. ...

For example you should mention that the mean and median became closer and spread out judging by the level value of standard deviation and thus by the graph. Because the mean and median (Chebyshev's theorem) matter some more when there are more outlier observations that were outliers (the zero values) and extreme observations.

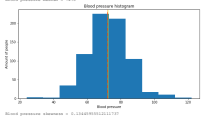
Also because you removed these zero values, which were in the left tail, you should mention that the mean and median values. That is also why the mean became bigger. Also, the % of the percentage of people with obesity (it's a relative percentage and now you have less people in the denominator of BMI>30, as those also were categorized as outliers, so it increased the percentage of the bmi).

```
BMI skewness = 0.5951874503963999
There is this smart percentage of people with obesity: 42.031410329241811
```

In [166]:
```
# 17.11. doing step 10 for Blood pressure
plt.hist(dataset.BloodPressure)
plt.title("Blood pressure histogram")
plt.xlabel("Blood pressure")
plt.ylabel("Amount of people")
bloodpressure_median = np.median(dataset.BloodPressure)
bloodpressure_mean = np.mean(dataset.BloodPressure)
print("Blood pressure median = " + str(bloodpressure_median))
plt.axvline(bloodpressure_median, color="red", linestyle="-")
plt.axvline(bloodpressure_mean, color="yellow", linestyle="-")
plt.xticks(ticks=[*Figure.figsize*] * [16, 9])
plt.show()
# ir distribution is skewed  on the mean more
# ir calculate skewness, it is positive and also very small, therefore tails are on the right, so we should sh
# skewness is pretty small here
bloodpressure_skewness = dataset.BloodPressure.skew()
print("Blood pressure skewness = " + str(bloodpressure_skewness))
# or obesity is not the thing here
```

Similar conclusions should have been done here.

```
Blood pressure median = 72.0
```



1 pt

Overall very good work. Conclusions need more work. GAD is not about making graphs, but rather being able to draw conclusions based on those graphs.

Total: 8.5 +0.1 to round up, so 9 out of 10
Good job!

```
Blood pressure skewness = 0.13424955512111737
```